SoftBase Db2 Tools

SoftBase
A Fresche Solutions Brand

Db2 Attach

Streamline Batch Processing

- **Reduce Batch Job Processing Time**
- **Eliminate Concurrency Problems and Deadlock Timeouts**
- **Run Batch Applications as Native z/OS Jobstreams**

Business Challenges

As companies and their Db2 databases continue to grow, database management becomes an increasingly complex task.

They face increasing pressure to grow revenues and at the same time strive to improve customer service. This demanding environment makes it difficult for many companies to complete their critical nightly batch processes on time.

Enhance Your Db2 Environment

SoftBase's Db2 Attach provides many features that enhance the batch Db2 environment, making it a valuable addition to the application developer's toolkit. Db2 Attach can reduce application development time by making developers more flexible in coding and deploying applications, and improve batch application reliability and efficiency without requiring source code changes.

Db2 Attach's patent-pending Multi-Row FETCH feature enables you to implement multi-row FETCH in existing applications without source code changes. Db2 Attach contains a patented Variable Commit Frequency (VCF) feature which can improve processing time by reducing an application's commit frequency. A LOCK feature and patented ENQ Serialization is included which improves application reliability by eliminating concurrency and deadlock timeouts. With deadlock timeouts eliminated, multithreading of Db2 batch jobs is possible, significantly increasing batch processing speed. Implementing Db2 Attach doesn't require any application source code changes; the batch application simply runs as a native z/OS program.

Reduce FETCH CPU Usage with MRF

The Db2 Attach MRF feature enables you to implement multi-row FETCH in your existing applications without requiring code changes. Db2 Attach intercepts your application's FETCH calls to Db2 as they are issued, and sends a modified multi-row FETCH call to Db2. DB2's result set is returned to a temporary table in the memory space. As your program issues subsequent FETCH calls, Db2 Attach intercepts them and quickly returns the row from its temporary table. MRF has demonstrated FETCH statement CPU time savings as high as 74%. temporary table.

Improve Batch Processing Time with VCF

Some applications are designed to commit at very frequent intervals to reduce contention issues. There may be times when contention is not an issue, and these unnecessary commits can be eliminated. Db2 Attach's Variable Commit Frequency feature allows users to reduce the number of commits issued to DB2 by an application, freeing up valuable system resources and saving time. Db2 Attach Monitoring feature is not intended to be a substitute for a Db2 performance monitor. Rather, it is a tool for application developers to use as a first level SQL performance analysis to identify possible problem areas in an application. Most application developers do not have easy access to Db2 performance information. In many cases, to obtain this information, they need the assistance of the Data Base Administrator. Users can adjust this interval throughout the day, based on performance needs. VCF has demonstrated CPU and Elapsed Time savings of between 17% and 60% for certain applications employing this feature. *Only Db2 Attach provides a dynamically variable commit ability.*

Reduce Batch Elapsed Time with Lock Feature

Db2 Attach provides an automatic means of locking tables. This feature will enable the application to issue Db2 table locks during execution without source code changes in the application. Because lock management is a significant portion of Db2 processing, utilizing Db2 Attach's LOCK feature for locking a table can reduce elapsed application run time in most installations.

Eliminate Concurrency Issues and DeadLock Timeouts

The patented ENQ serialization feature in the Db2 Attach solves concurrency problems with your Db2 batch programs without the need for elaborate scheduling. These problems typically arise when multiple programs are simultaneously attempting to update the same table spaces while performing commits infrequently. The ENQ Serialization feature provides a mechanism to force the serial execution of these programs that update the same tables and thereby eliminate the possibility of deadlock timeouts and resource unavailable conditions.

Improve Batch Throughput with Multi-Threading

The ENQ Serialization feature provides the ability to multi-thread batch DB2 job streams. The alternatives are to single-thread all batch DB2 processes, or control access to DB2 with a z/OS scheduling package. Using these alternatives are possible when workloads are small; however, as workloads increase and batch processing windows shrink, this strategy becomes difficult to adhere to without adding more processing power. Db2 Attach can increase batch throughput and potentially delay a processor upgrade.

intercepts commits issued by an application, counts these commits, and suppresses them until the count reaches a number specified by the user

SQL Performance Monitoring Feature

The SQL Monitoring feature provides a method for recording performance information about the execution of SQL statements from a batch application program. When invoked, the SQL Monitoring feature will record CPU time, duration, CPU percent and other pertinent information about each SQL statement's performance. The SQL

Additional Features and Benefits

Automatically formats output SQL error messages, SBAIN/SYSTSIN parameters, and error messages about those parameters

- Provides an exit program for overriding system/plan default
- Allows the user to perform a TORBA recovery in the event of an ABEND after one or more COMMITs have been performed
- Allows the application to perform CONNECT, OPEN, CLOSE and DISCONNECT functions, giving it complete control to Call Attach
- No source program changes are required to use Db2 Attach
- Can be installed in a few minutes by a nontechnical person

JCL Example: BEFORE with TSO Batch – and – AFTER using Db2 Attach

A conventional Db2 TSO batch job step is displayed followed by the same step after converting to use Db2 Attach. Batch Db2 jobs can look and act like normal z/OS batch jobs using Db2 Attach. Parameters may be passed on the EXEC statement whereas batch TSO must input parms through SYSTSIN. JCL scan utilities can now easily identify the name of the program executing in each step. Return codes generated by DB2 programs can be passed to subsequent job steps. There is no need for a RETRY parameter to handle Db2 outages. Db2 Attach automatically waits until Db2 becomes active again.

Sample JCL not using Db2 Attach

```
//PLAINCOB EXEC PGM=IKJEFT01,REGION=0M      ←for batch TSO, program is IKJEFTxx
//STEPLIB  DD DISP=SHR,DSN=DSN910.SDSNLOAD ←DSNLOAD may be first in concatenation
//         DD DISP=SHR,DSN=USER.TEST.LOAD
//INFILE1 DD DSN=USER.TEST.INFILE,DISP=SHR
//OUTDD1   DD DSN=USER.TEST.OUTFILE,DISP=(MOD,CATLG,CATLG),
//         UNIT=3390,SPACE=(TRK,(2,2)),
//         DCB=(RECFM=FB,LRECL=615,BLKSIZE=27675)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *                              ← TSO Run time parms
 DSN SYSTEM(DSN1) RETRY(5)
 RUN PROGRAM(PLAINCOB) PLAN(PLAINCOB) PARM('TESTING,MARCH ')  ←app parms
 END
//
```

After **PLAINCOB** is prepared to use Db2 Attach the execution JCL must be changed to reference the application program name in the 'EXEC PGM=' statement and the STEPLIB concatenation is changed with the Db2 Attach load library ahead of SDSNLOAD. The following sample JCL represents these changes. Note the following JCL statements:

- PGM=PLAINCOB in the execute statement
- The program parameters are now included in the PARM= field in the execute statement
- The STEPLIB concatenation order, SBS.SDT110.LOAD is ahead of SDSNLOAD
- The addition of SBAPRINT and SQLMSGS DD statements
- The replacement of SYSTSIN of an SBAIN DD statement and the instream Db2 Attach parameters

Sample JCL using Db2 Attach

```
//PLAINCOB EXEC PGM=PLAINCOB,PARM='TESTING,MARCH '  ←User pgm name,app parms
//STEPLIB  DD DISP=SHR,DSN=SBS.SDT110.LOAD  ←list SoftBase Db2 Tools first in library concatenation
//         DD DISP=SHR,DSN=DB2.SDSNLOAD
//         DD DISP=SHR,DSN=USER.TEST.LOAD
//SBAIN    DD  *                            ←SBAIN DD replaces SYSTSIN DD
 SYSTEM(DSN1)
 PLAN(PLAINCOB)
//INDD1   DD DSN=USER.TEST.INFILE,DISP=SHR
//OUTDD1   DD DSN=USER.TEST.OUTFILE,DISP=(MOD,CATLG,CATLG),
//         UNIT=3390,SPACE=(TRK,(2,2)),
//         DCB=(RECFM=FB,LRECL=615,BLKSIZE=27675)
//SYSPRINT DD SYSOUT=*
//SBAPRINT DD SYSOUT=*
//SQLMSGS  DD SYSOUT=*
```

SoftBase Db2 Tools

Db2 Attach's MRF Feature

SoftBase
A Fresche Solutions Brand

Multi-Row Fetch Feature

- **Implement multi-row FETCH automatically NO RECODING REQUIRED!**
- **Significant CPU savings gives pause to expensive CPU Upgrade**
- **Quickly determine if MRF will benefit a given cursor without recoding the application**

Business Challenges

The SoftBase **Db2 Attach** MRF solution can cut your DB2 FETCH statement CPU time.

Are you taking advantage of Db2's multi-row FETCH?

As a database professional, you know that reducing CPU processing and elapsed time translates directly into time and money saved. For large organizations, this can translate into millions of dollars in mainframe costs saved annually.

But there are several challenges to implementing multi-row Fetch in legacy Db2 applications

- Finding applications that can benefit from multi-row **FETCH is DIFFICULT**
- Developers **SPEND TIME** learning required coding techniques
- Recoding is **TEDIOUS** and **RISKS DAMAGING PERFORMANCE**

Implementing can require **MONTHS OF TESTING** to ensure that the application's performance isn't compromised.

SoftBase's MRF solution makes it easy!

SoftBase's patent-pending Multi-Row FETCH (MRF) feature of its Attach is a powerful tool for automatically implementing multi-row FETCH in your existing batch Db2 applications no application coding changes required! Many applications are I/O bound and MRF can in certain cases make that worse instead of better. The Db2 Attach MRF feature can be quickly used to investigate whether a given cursor is worth the effort to retro-fit your application.

With MRF, you can enjoy the performance improvements that natively coded multi-row fetch provides batch applications without tedious recoding and testing. In recent tests, the MRF feature demonstrated FETCH statement CPU time savings.

What can MRF mean for you?

- Bypassing months of coding and testing for **HUGE PRODUCTIVITY SAVINGS**
- Delaying a CPU upgrade for **HUGE MAINFRAME COST SAVINGS**
- Reducing risk of delaying critical applications **PREVENTS LOST REVENUE**

And, by using **SoftBase's Db2 Batch Analyzer's** free trial, you can estimate performance improvement before implementing **Db2 Attach's MRF** feature.

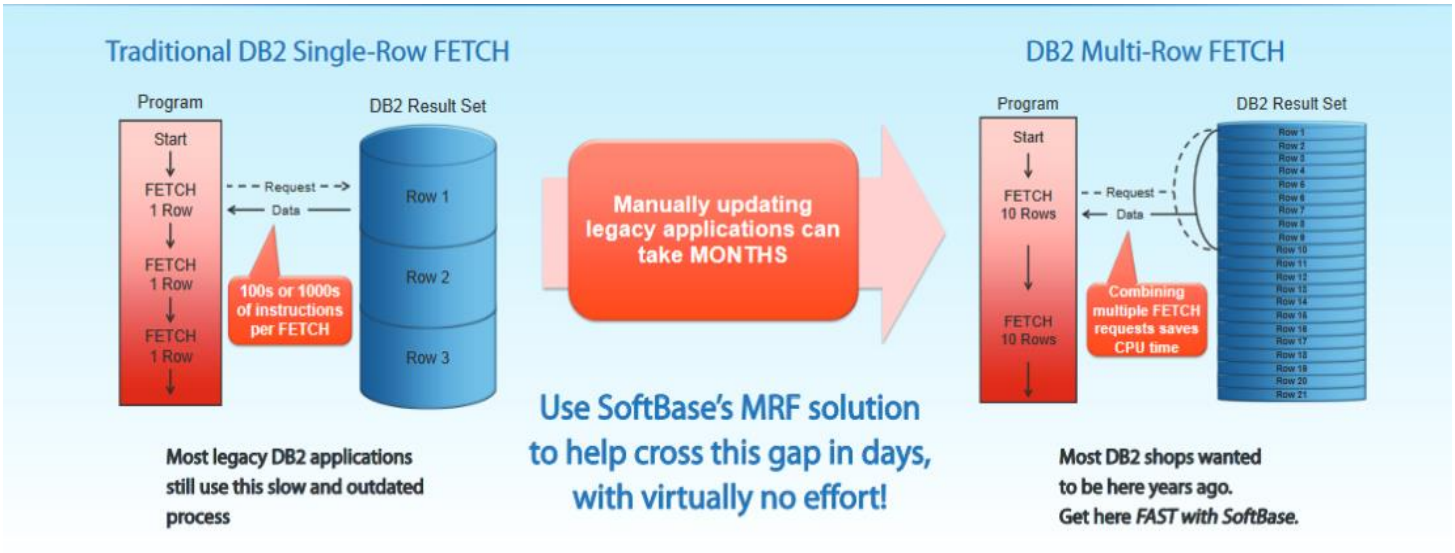
No commitment. Instant gratification. See how it works……..

For Information: Call: 800.669.7076 | E-mail: sbsales@freschesolutions.com **| Visit:** www.softbase.com

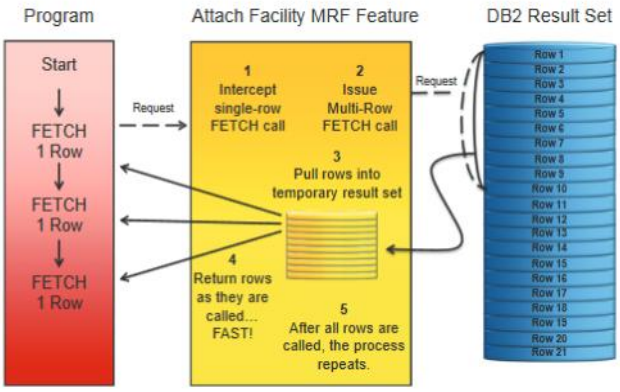
SoftBase Db2 Tools

SoftBase
A Fresche Solutions Brand

Db2 Attach's MRF Feature



How does Attach Facility's MRF feature work?

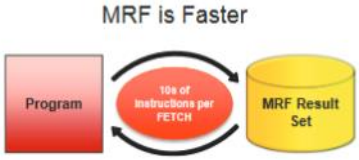


The MRF feature for Attach Facility implements Multi-Row FETCH functionality in legacy applications **WITHOUT SOURCE CODE CHANGES.**

How does Attach Facility's MRF feature save CPU?

Attach Facility's MRF Feature can reduce the number of CPU instructions per FETCH call, reducing overall CPU usage.

As each FETCH call issued using MRF requires only tens of instructions, versus hundreds or thousands when issued to DB2 via single-row FETCH, each row can be pulled much faster, reducing CPU usage and elapsed time.



To determine if SoftBase's MRF solution is a fit for you, contact SoftBase today.